

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

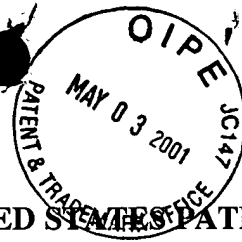
Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**



0400  
5-4-01

Docket No. 121.1012

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: )  
Kenichiro SAKAI et al. )  
Serial No.: 09/835,625 )  
Filed: April 17, 2001 )

Group Art Unit: *To be Assigned*  
Examiner: *To be Assigned*

For: TWO-DIMENSIONAL CODE EXTRACTING METHOD

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN  
APPLICATION IN ACCORDANCE  
WITH THE REQUIREMENTS OF 37 C.F.R. §1.55**

*Assistant Commissioner for Patents  
Washington, D.C. 20231*


*Sir:*

In accordance with the provisions of 37 C.F.R. §1.55, the applicants submit herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2000-116429  
Filed: April 18, 2000

It is respectfully requested that the applicants be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,  
STAAS & HALSEY LLP

By:   
Gene M. Garner II  
Registration No. 34,172

700 11th Street, N.W., Ste. 500  
Washington, D.C. 20001  
(202) 434-1500  
Date: May 3, 2001



日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2000年 4月18日

出 願 番 号

Application Number:

特願2000-116429

出 願 人

Applicant(s):

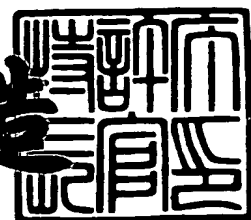
富士通株式会社



2001年 4月20日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3032489

【書類名】 特許願

【整理番号】 0050462

【提出日】 平成12年 4月18日

【あて先】 特許庁長官殿

【国際特許分類】 H04N 1/403

【発明の名称】 二次元コード抽出方法

【請求項の数】 7

【発明者】

    【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

    【氏名】 酒井 憲一郎

【発明者】

    【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

    【氏名】 野田 嗣男

【特許出願人】

    【識別番号】 000005223

    【氏名又は名称】 富士通株式会社

【代理人】

    【識別番号】 100108202

    【弁理士】

    【氏名又は名称】 野澤 裕

    【電話番号】 044-754-3035

【手数料の表示】

    【予納台帳番号】 011280

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

特 2 0 0 0 - 1 1 6 4 2 9

【物件名】            要約書    1

【包括委任状番号】    9913421

【プルーフの要否】    要

【書類名】 明細書

【発明の名称】 二次元コード抽出方法

【特許請求の範囲】

【請求項 1】

入力した画像データを  $M \times N$  画素 ( $M$ 、 $N$  は正の整数) の矩形ブロック単位に走査し、

前記走査したブロックから所定の条件を満たすブロックを検出し、

前記検出されたブロックの中で隣接して連結しているブロックにより構成される領域を検出し、

前記連結領域の検出に基づいて二次元コードの領域を検出し、前記入力画像から二次元コードの画像を抽出することを特徴とする、二次元コード抽出方法。

【請求項 2】

請求項 1 記載の二次元コード抽出方法において、

前記ブロックの検出は、ブロック内の白画素と黒画素の比率が所定の範囲内にあるブロックを、二次元コードを構成するブロックとして検出することを特徴とする二次元コード抽出方法。

【請求項 3】

請求項 1 記載の二次元コード抽出方法において、

前記ブロックの検出は、ブロック内の総画素数に対するブロック内の水平ラインおよび／または垂直ライン内の画素値の変化点の個数の割合が所定の範囲内になるブロックを、二次元コードを構成するブロックとして検出することを特徴とする二次元コード抽出方法。

【請求項 4】

請求項 1 記載の二次元コード抽出方法において、

前記ブロックの検出は、ブロック内の各ラインに含まれる黒画素の水平方向の射影または／および垂直方向の射影の変化の度合いが所定の範囲内になるブロックを検出することを特徴とする二次元コード抽出方法。

【請求項 5】

請求項 1 記載の二次元コード抽出方法において、

前記検出された連結領域の中で当該領域を構成するブロック数が所定の範囲内の領域を選択し、該選択した領域を囲む領域を二次元コード領域として検出することを特徴とする二次元コード抽出方法。

【請求項 6】

請求項 1 記載の二次元コード抽出方法において、

前記検出された二次元コードの領域内における任意の位置から、上下方向および左右方向に所定の大きさの矩形ブロック単位に走査し、

走査した矩形内の黒画素数が所定の値以下になる位置を各方向について検出し、

検出位置を囲む矩形領域を二次元コードとして抽出することを特徴とする二次元コード抽出方法。

【請求項 7】

請求項 1 記載の二次元コード抽出方法において、

前記走査したブロック内に存在する黒画素同士の平均距離を求め、

前記求められた黒画素間平均距離の値が所定値以上の場合に、当該ブロックを二次元コードを構成するブロックとして抽出することを特徴とする、二次元コード抽出方法。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明は、文書や図表とともに二次元コードが印刷された紙面をスキャナ等の画像入力装置により読み取った画像データから、二次元コードの画像を自動的に抽出する二次元コード抽出方法に関する。

【 0 0 0 2 】

【従来の技術】

近年、従来のバーコードよりも多くの情報を埋め込むことができる二次元コードの開発が進み、実用化され始めている。二次元コードを利用することにより、文書や画像データ、音声データなどのあらゆる種類のバイナリデータを二次元コードとして紙面に印刷することが可能になった。

【 0 0 0 3 】

二次元コードを印刷した紙は、ファクシミリで伝送したり印刷物として配布する。受け取り側では、紙面の二次元コードをイメージスキャナにより画像データとして読み取って、各二次元コードの復元に対応したソフトウェアやハードウェアにより元のデータに復元することができる。

【 0 0 0 4 】

二次元コードの中には一般的なイメージスキャナでの読み取りを可能にするものもあり、専用の読み取り装置が無くても手持ちのスキャナにより読み取って元のデータを復元することができるため、読み取り手段の汎用性が高いというメリットがある。

【 0 0 0 5 】

【発明が解決しようとする課題】

二次元コードだけを紙面に印刷するのではなく、文書や図表とともに任意の位置に任意の大きさの二次元コードを紙面に印刷した場合、汎用のイメージスキャナを使って二次元コード部分の読み取りを行うと二次元コードに加えて文字や図表の一部も画像データとして取り込まれる可能性が非常に高い。

【 0 0 0 6 】

このように二次元コード以外の画像が入っていると、二次コード復元ソフトウェアやハードウェアで正常に復元することができないため、イメージスキャナで読み取った画像データから二次元コード部分だけを抽出しなければならないという課題があった、

【 0 0 0 7 】

【課題を解決するための手段】

そして、この目的は、入力した画像データを $M \times N$ 画素（ $M$ 、 $N$ は正の整数）の矩形のブロック単位に走査し、走査したブロックから所定の条件を満たすブロックを検出し、検出されたブロックの位置に基づいて二次元コードの位置を検出して入力画像から二次元コードの画像を抽出する二次元コード抽出方法により達成される。

【 0 0 0 8 】



また、ブロック検出は、ブロック内の白画素／背景色と黒画素／文字色の比率が所定の範囲内になるブロックを検出するようにしてもよい。

【 0 0 0 9 】

また、ブロック内の総画素数に対するブロック内の水平ラインおよび／または垂直ライン内の画素値の変化点の個数の割合が所定の範囲内になるブロックを検出する、ブロック内の黒画素の連結の仕方に基づいてブロックを検出する、あるいはブロック内の各ラインに含まれる黒画素の水平方向の射影または／および垂直方向の射影の変化の度合いが所定の範囲内になるブロックを検出するようにしてもよいし、これらを組み合わせてブロックを検出してもよい。

【 0 0 1 0 】

一方、二次元コードの検出の際には、検出された連結領域の中でその領域を構成するブロック数が最大の領域を選択し、該領域を囲む矩形領域を二次元コード領域として検出するようにしてもよい。

【 0 0 1 1 】

更に、連結領域の中で領域を構成するブロック数が所定の範囲内の領域を選択し、選択されたそれぞれを囲む矩形領域を二次元コード領域として検出するようにしてもよい。

【 0 0 1 2 】

また、二次元コード抽出に際して、二次元コード領域内における任意の位置から垂直上下方向および水平左右方向に所定の大きさの矩形ブロック単位に走査し、走査した矩形内の黒画素数が所定の値以下になる位置を各方向について検出して検出位置を囲む矩形領域を二次元コードとして抽出するようにしてもよい。

【 0 0 1 3 】

また更に、二次元コード領域の周囲に所定幅の空白を付与して抽出するようにしてもよい。

【 0 0 1 4 】

一方、二次元コード抽出にあたって二次元コードの傾きを検出し、傾きを補正しながら二次元コード画像を抽出するようにしてもよい。

【 0 0 1 5 】

その際に、二次元コードの上下端および／または左右端それぞれの境界上における複数箇所の座標値を直線近似して、二次元コードの傾き角度を検出するようにすればよい。

【 0 0 1 6 】

また、入力画像の読取方向を検出して正立像に補正することによって、入力画像の向きを正しく補正することが可能となる。

【 0 0 1 7 】

【発明の実施の形態】

以下、図面を用いて本発明の実施形態を説明する。

【 0 0 1 8 】

一般的に、二次元コードはバイナリデータを特定のアルゴリズムにより画素パターン変換したものであり、文書や図等と異なり画像としての見た目の意味を持たない。一方、画像レベルで見ると、文書や図表を構成する画素のパターンと二次元コードを構成する画素パターンの特徴は異なっている。この相違は見た目にも明らかであり、文書・図表と二次元コードが混在している画像から二次元コード部分を目視により容易に識別できる。

【 0 0 1 9 】

図 1 はイメージスキャナで読み取った、二次元コードと文書や図表が混在する文書画像を表しており、横 M 画素、縦 N 画素（いずれも正の整数、 $M = N$ であってもよい）のブロック毎に読み取られた画像を走査した例が図示されている。図中、走査された画像中のブロックは破線で囲まれた矩形で図示されている。また、実線で囲まれた矩形は、後に詳細を説明する、本実施形態により抽出されたブロック（特に二次元コードを構成すると判別されたブロック）を示している。なお、図 1 において二次元コードは画像の右下の矩形の領域に相当し、また、検出されたブロック以外の文書や図表などは、特に図中には明示していない。

【 0 0 2 0 】

なお、以降の説明では文字や図表および二次元コードを構成する画素を「黒画素」、その背景となる画素を「白画素」とする。

【 0 0 2 1 】

本実施形態では、文書・図表と二次元コードの画素パターンが相違する点に着目した。まず、画像をブロック単位に走査してブロックごとに二次元コード（あるいは二次元コードを構成するブロックと推測される部分）か否かを検出し、二次元コードの存在確率が高い部分を探し出す

ブロックが二次元コードの一部か否かを判断するためには、ブロック内の黒画素分布を求める。これは、二次元コード部分と文字領域とで相違することを利用したものである。

#### （１）ブロック内の黒画素率により識別

この方法では、二次元コードの黒画素率がブロック単位で見るとほぼ一定であることを利用する。この際に、ノイズやスキャナ特性による変動を考慮し、二次元コードと判断すべきブロック内の黒画素率範囲を予め定めておく。この範囲は、あらかじめ二次元コードを走査して、その結果に基づいて設定するようにしてもよい。走査したブロックの黒画素率がこの設定した黒画素率の範囲内にあれば、そのブロックが二次元コードを構成するブロックであると判断する。二次元コードを構成していないブロックについては、黒画素率が二次元コードを構成するブロックの範囲には入らないため、上記の方法を用いることにより、二次元コード以外のブロックの多くを取り除くことができる。

#### （２）白画素と黒画素との変化点の割合により識別

この方法では、二次元コードの画素の変化パターンがブロック単位で見るとほぼ一様であることを利用する。ブロック内を水平または垂直に１ライン単位に走査し、各ライン内の画素の変化点、すなわち黒画素から白画素、または白画素から黒画素に変化した回数を計数し、ブロック全ラインについて求めた変化点の総数をブロックの総画素数で割った値を「変化点の割合」とする。二次元コードブロックの変化点の割合を予め調べておき、ノイズやスキャナ特性による変動を考慮して二次元コードと判断する変化点の割合の範囲を定める。

#### 【 0 0 2 2 】

走査したブロックの変化点の割合が、この範囲内にあれば二次元コードを構成するブロックと判断する。これにより、二次元コード以外のブロックの多くが取り除かれる。

### (3) ブロック内の黒画素の連結の仕方で識別

二次元コード内の黒画素はブロック内ではほぼ一様に分布するのに対し、文字を含むブロックでは黒画素により文字が構成されるため、黒画素が局所的にかたまって存在している。この方法では、上記した画像の特徴を二次元コード検出に利用する。

### (4) ブロック内の黒画素の射影の変動から識別

変化点の場合と同様に、二次元コード内の画素分布がほぼ一様である。ブロックを水平または垂直に1ライン単位に走査し、各ラインに含まれる黒画素数を計数することで該ブロックの射影を得る。二次元コードのブロックの場合には、画素分布がほぼ一様であることからライン間の射影の変化が小さく、文書や図表の場合には、黒画素分布が一様でないことからライン間で射影の変化が大きくなる。

#### 【0023】

従ってあらかじめ二次元コードと判断する射影の変動の範囲を定めておき、走査したブロックの射影の変動がこの範囲内であれば二次元コードのブロックと判断する。これにより、二次元コード以外のブロックの多くを取り除くことができる。

#### 【0024】

二次元コードブロックを検出する際に、上記(1)～(4)の中から2つ以上の方法を組み合わせることで、二次元コードブロックの検出精度を大幅に向上させることができる。

#### 【0025】

なお、上記の黒画素分布を利用して二次元コードを構成するブロックを検出したとしても、二次元コードとして検出されたブロックの中には、二次元コード以外の画像が含まれる可能性がある。これは、一つのブロック単位でみた場合、そのブロックが二次元コードではないにもかかわらず、上記の黒画素分布の条件を満たしてしまうことがあるためである。しかし、これらの二次元コードを構成していない、言わば誤検出されたブロックは、自分自身の周辺のブロックが二次元コードとして検出される条件を満たしていない可能性が高いと考えられる。その

ため、このようなブロックに関しては、二次元コードとして検出されたブロックが他のブロックから孤立して存在したり（図 1 A）、数個程度のブロックが隣接して検出される（図 1 B、C）に過ぎない。

【0 0 2 6】

一方、二次元コード領域内においては、二次元コードとして検出されるブロックの数が二次元コード領域外に比べて多くなるため、検出されたブロックが図 1 の D のように多く隣接して繋がっている。

【0 0 2 7】

そこで、二次元コードとして検出されたブロックのうち、隣接して連結するブロックの個数が最大となる領域を検出し、その領域の上下左右に所定の幅を持たせた矩形で囲み、この領域を検出することができる。

【0 0 2 8】

ここで、上記した所定の幅を最大ブロックの幅とすることで、常に二次元コード全体を含んだ領域が検出される。

【0 0 2 9】

また、連結するブロックが所定の個数以上の領域を選択することで、同一画像内に複数個の二次元コードが存在する場合でも抽出することが可能となる。

【0 0 3 0】

さらに、以下の方法により、より正確に二次元コードの領域を検出することができる。

【0 0 3 1】

図 2 は、前述した方法により二次元コードとして検出された、隣接ブロックの個数が最大となる領域を拡大して示した図である。図 2 において、一番外側の実線で示した矩形内（矩形領域 1）が検出する二次元コードの領域、その内側の矩形（矩形領域 2）が二次元コードとして検出されたブロックに外接する矩形領域である。また、一番外側の破線と矩形領域 1 の間の領域は、二次元コードと文字や図表を隔てるために設けられた、全ての画素が白画素の空白領域である。

【0 0 3 2】

まず、矩形領域 2 内に、基準位置を設定する。図 2 では基準位置を丸印で示し

ており、検出された領域の中心に設定された例が図示される。この基準位置を中心として画像に対して左右の方向、上下方向に向かって二次元コードの境界を検出する処理を行う。二次元コードの境界から外側には、所定の幅の空白領域が付与されるのが一般的である。従って、検出基準位置から二次元コードの縁に向かってブロック単位に走査して行き、走査したブロック内の全面素が白画素になった時点で上記した空白領域に達したことが判り、このブロックの基準位置側の辺が二次元コードの境界ということになる。

#### 【 0 0 3 3 】

ここで、矩形領域 2 の内側は二次元コード領域とみなせるため、二次元コードの境界の検出は、図 2 の A ～ D のように矩形領域 2 の外側から開始すればよい。各検出開始位置 A ～ D から二次元コードの境界方向に向かってブロック単位に上記の検出を行えば、その内側の全面素が白画素になるブロック A' ～ B' が検出される。そして、A' ～ B' の検出基準位置側（すなわち二次元コード側）の辺を通る矩形が二次元コードの矩形領域として検出される。

#### 【 0 0 3 4 】

上記のようにして検出した二次元コードを抽出する際に、二次元コードを復元するソフトウェアやハードウェアが必要とする所定の幅の白画素を二次元コードの周囲に付与することで、正しく復元できる二次元コードの画像にすることができる。

#### 【 0 0 3 5 】

ハンドヘルド型のイメージスキャナにより紙面上の二次元コード部分の読み取りを行った場合、手動でスキャナを操作するためスキャナが斜めに動き読み取った画像に傾が生じる場合が考えられる。

#### 【 0 0 3 6 】

そこで、二次元コードを抽出する際に同時に二次元コードの傾き角度も検出し、二次元コード抽出時に傾きを補正しながら抽出を行うことで、傾きのない二次元コードを抽出することができる。

#### 【 0 0 3 7 】

二次元コードが矩形である場合には、二次元コード上下の辺、すなわち二次元

コードの境界線が平行であることから、二次元コードを検出する際に上下の境界線上の点を複数箇所検出し、画像内におけるそれらの座標値からこれらの点を直線近似することで境界線の傾きを求め、それを二次元コードの傾き角度とすることができる。左右の辺についても互いに平行であるため、同様にして傾き角度を検出することができ、上下左右から傾きを検出することで傾き角度の検出精度が高まる。

## 【 0 0 3 8 】

手動操作型のハンドヘルドスキャナにより読み取る場合、ハンドヘルドスキャナは上下左右いずれの方向にもスキャンできることから、読み取った画像が回転したり鏡像反転して入力される場合がある。二次元コードの検出処理を始める前に読み取り方向を検出して正立像に補正しておいたり、二次元コードを検出した後抽出する際に正立像になるように補正しながら抽出することで、常に正しい向きの二次元コードを抽出することができる。

## 【 0 0 3 9 】

以下、図面に基づいて本発明の実施例について詳細に説明する。

## 【 0 0 4 0 】

図 3 は本発明の全体の処理フローである。まず、入力画像をブロック単位に走査して、上記した (1) ～ (4) の方法を用いて、所定の条件を満たすブロックを検出する (S 1 0 1)。次に、検出されたブロックが連結している領域を検出する (S 1 0 2)。続いて、S 1 0 2 により検出された領域に基づいて、二次元コードの領域を検出する (S 1 0 3)。そして、検出した二次元コード領域から二次元コード画像を抽出する (S 1 0 4)。最後に、二次元コードの傾き角度を検出し、必要ならば傾き補正を行なう (S 1 0 5)。

## 【 0 0 4 1 】

図 4 は、入力画像から検出したブロックの情報を格納するためのブロック情報テーブルの構成を示す図である。ブロック番号は二次元コードの一部として検出されたブロックに、検出した順番につける一連番号である。検出されたブロックの最後を示すために、テーブル最後に E O B (End Of Block table) を格納する。

## 【 0 0 4 2 】

検出されたブロックごとに、ブロック左上のX座標とY座標、そのブロックに結合するブロック数、結合するブロックにより構成される領域の一連番号が格納される。検出されたi番目のブロックの、ブロック情報テーブルの各項目の値の表記は、X座標を $X_i$ 、Y座標を $Y_i$ 、結合ブロック数を $C_i$ 、領域番号を $R_i$ で表すこととする。

## 【 0 0 4 3 】

以下の実施例では、本ブロック情報テーブルを使用して二次元コードの検出を行なう。

## 【 0 0 4 4 】

図5は二次元コードのブロックを検出する処理フローを示す図であり、図3のS101の処理に対応する。図中の記号は、 $x$ と $y$ は走査するブロック左上のそれぞれX座標とY座標、 $M$ と $N$ はそれぞれブロックの横画素数と縦画素数を表す。また、WIDTHとHEIGHTは、それぞれ入力画像の横画素数と縦画素数を表す。 $i$ は検出したブロックの番号を示すための変数である。

## 【 0 0 4 5 】

まず初期設定として、 $x = 0$ 、 $y = 0$ すなわち入力画像の右上の原点をブロックの左上の座標に設定し、同時に最初のブロックを示すよう $i = 1$ に設定する（S102）。

## 【 0 0 4 6 】

S202では、走査するブロックを左上の座標を $(x, y)$ とする幅 $M$ 画素、高さ $N$ 画素の矩形領域に設定し、このブロックが所定の条件、すなわち黒画素率や変化点の個数、黒画素の連結状態が所定の値であるか否かを識別する（S203）。S203において、該ブロックが所定の条件を満たさないと判断した場合には処理をS206に移し、条件を満たすと判断した場合には処理をS204に進める。

## 【 0 0 4 7 】

S204では、ブロック情報テーブルの*i*番目の項目を選択し、 $X_i$ に $x$ 、 $Y_i$ に $y$ をそれぞれ格納し、次に検出されるブロックを示すようにブロック番号*i*に1を加算する（S205）。



## 【0048】

S206では、 $x$ に $M$ を加えることで、次に走査するブロック左上の座標を $X$ 軸方向に $M$ 画素分移動させる。S207では、 $x+M$ が $WIDTH$ 以上か否か、すなわち $x$ を左上の $X$ 座標とする幅 $M$ 画素のブロックの右端が、入力画像の幅を超えていないか否かをチェックする。 $x+M$ が $WIDTH$ 未満の場合にはS202に処理を戻す。 $x+M$ が $WIDTH$ 以上の場合には、S208で $x=0$ すなわち走査するブロックの $X$ 座標を入力画像の左端に設定する。

## 【0049】

S209では、 $y$ に $N$ を加えることで、次に走査するブロック左上の座標を $Y$ 軸方向に $N$ 画素分移動させる。S210では、 $y+N$ が $HEIGHT$ 以上か否か、すなわち $y$ を左上の $Y$ 座標とする高さ $N$ 画素のブロックの下端が、入力画像の高さを超えていないか否かをチェックする。 $y+N$ が $HEIGHT$ 以上の場合にはS211に処理を移す。S211ではブロック情報テーブルの $i$ 番目のブロック番号に $EOB$ を格納して処理を終了する。

## 【0050】

一方、S210において $y+N$ が $HEIGHT$ 未満の場合にはS202に処理を戻す。

## 【0051】

S202からS210までを繰り返し行なうことにより、入力画像全体をブロック単位に走査して、図7の太線で囲ったブロック1～26のように、二次元コードと識別したブロックを検出し、それら各ブロックの $X$ 座標および $Y$ 座標が図8のブロック情報テーブル1～26に示したように検出される。

## 【0052】

なお、図7はブロックの幅 $M=32$ 画素、高さ $=32$ 画素の場合の例である。図6は、図3のS101により検出したブロックが連結して構成する領域を検出する処理フローを示す図であり、図3のS102の処理に対応する。

## 【0053】

図中の記号は、 $i$ はブロック情報テーブル内におけるブロック番号を示すための変数、 $k$ は検出した領域の番号を示すための変数である。

## 【0054】

まず初期設定として、ブロック情報テーブル（図 4）の先頭から EOB までの結合ブロック数と領域番号を全て 0 でクリアしておく（S301）。

## 【 0 0 5 5 】

S302 ではブロック番号  $i = 1$  すなわちブロック情報テーブルの先頭を指し示すように設定し、領域番号を  $k = 1$  すなわち最初の領域を示す値を設定する。

## 【 0 0 5 6 】

S303 では、 $i$  番目のブロックテーブルの領域番号  $R_i$  が 0 か否か、すなわち  $i$  番目のブロックが既に何らかの領域に属しているか否かをチェックし、0 ならば領域に属していないと判断して処理を S304 に移し、0 でなければ既に領域に属していると判断して S307 に処理を移す。

## 【 0 0 5 7 】

S304 では、例えば「境界追跡法」などの方法によりブロック  $i$  を始点とするブロックにより構成される領域を検出し、該検出した領域を構成するブロックの個数  $P$  を求める。なお、境界追跡法は一般的に知られている方法であり、例えば「画像処理」（尾崎他著、共立出版、1993、p187～188）や「画像処理工学」（谷口・編、共立出版、1996、p116～118）に記載されている。

## 【 0 0 5 8 】

S305 では、ブロック情報テーブルを参照し、S301 で検出した領域に含まれる全てのブロック数  $C_i$  に  $P$ 、領域番号  $R_i$  に  $k$  を設定する。

## 【 0 0 5 9 】

S306 では、領域番号  $k$  に 1 を加える。

## 【 0 0 6 0 】

S307 では、次のブロック情報テーブルの項目を参照するために  $i$  に 1 を加える。

## 【 0 0 6 1 】

S308 では、 $i$  番目のブロック情報テーブルのブロック番号欄が EOB か否か、すなわち S101 で検出した全てのブロックについて処理を終えたか否かチェックする。ブロック番号が EOB でない場合には、処理を S303 に戻し、ブロック番号が EOB である場合には全処理を終えたと判断して修了する。

## 【 0 0 6 2 】

S303からS308までの処理を繰り返し行なうことで、図7の領域1～4のように二次元コードと識別されたブロックが連結して構成する領域が検出され、図8のブロック情報テーブルのように各ブロックについて結合ブロック数と領域番号が求まる。例えば、ブロック6の場合には、結合するブロック数が20個の領域に所属し、領域番号が3となる。

## 【 0 0 6 3 】

図9は二次元コード領域を検出する処理の第一の実施例を示す処理フローであり、図3のS103の処理に対応する。

## 【 0 0 6 4 】

S401では、ブロック情報テーブルを参照し、結合ブロック数の最大値を検出してRmaxとする。図8の例では、Rmax=20 となる。

## 【 0 0 6 5 】

S402では、結合ブロック数がRmaxに一致するブロックの中で、X 座標およびY 座標の最小値 (Xmin、Ymin) と最大値 (Xmax、Ymax) を求める。図8の例では、座標の最小値 (Xmin、Ymin) としてブロック6の座標 (X6、Y6) = (192、352)、座標の最大値 (Xmax、Ymax) としてブロック26の座標 (X26、Y26) = (352、448) が検出される。

## 【 0 0 6 6 】

S403では、左上角の座標を (Xmin-M、Ymin-N)、右下角の座標を (Xmax+2×M、Ymax+2×N) とする矩形領域を二次元コード領域として決定する。図7の例では、図10Aの領域3から一ブロック分離れた矩形領域C が二次元コード領域として検出される。

## 【 0 0 6 7 】

本方法では、実際の二次元コードである図10Bに対して、大きめの領域が二次元コードとして検出されることになる。

## 【 0 0 6 8 】

図11は、二次元コード領域を検出する処理の第二の実施例を示す処理フローあり、図3のS103の処理に対応する。

## 【 0 0 6 9 】

S501では、ブロック情報テーブルを参照し、結合ブロック数の最大値を検出してRmaxとする。図8の例では、Rmax=20 となる。

## 【 0 0 7 0 】

S502では、結合ブロック数がRmaxの領域の中心を二次元コード領域検出の基準位置に決定する。図8の例では、図10のように結合ブロック数が最大の領域として領域3（図10A）が選択され、その中心として図12Dが決定される。ここでの中心位置は、領域3に外接する矩形領域の中心で良い。

## 【 0 0 7 1 】

S503では、S502で決定した検出基準位置から、上下および左右の方向に矩形のブロック単位に走査し、該ブロック内の黒画素の個数がQ 個以下になる位置を検出する。Q は0にしても良いが、ノイズなどの影響を考慮してある程度の個数を設定しておいても良い。

## 【 0 0 7 2 】

図12は、図10の領域3部分を拡大した図である。この例では、それぞれ左側でA、右側でB、上側でC、下側でDの位置が検出される。Q=0に設定し、ノイズがない場合には、検出したA付近は図12下図のようになり、A' の位置すなわちブロックの基準位置側を検出した位置に決定すればよい。

## 【 0 0 7 3 】

S504では、S503で検出した上下左右の四点から、検出基準位置と反対方向に所定の画素数分はなれるように矩形を二次元コードの領域として決定する。図12では、A～Dの各検出位置からE画素分空けた位置を通るように設定した矩形（一番外側の矩形）が二次元コード領域となる。

## 【 0 0 7 4 】

以上、二次元コード領域を検出する第一および第二の実施例では、二次元コードが傾いている場合でも、二次元コード全体を含む領域を検出することができる。更に、検出した二次元コードの傾きを検出し、傾きを補正することでより正しく二次元コードを復元できる。

## 【 0 0 7 5 】

図 1 3 は、二次元コードの傾き補正の処理フローであり、図 3 の S105 の処理に対応する。

#### 【 0 0 7 6 】

S601 では、二次元コード内の検出基準位置から、上下左右に所定の間隔でブロック単位に走査し、ブロック内の黒画素が Q 個以下になる位置を検出する。ここでの検出方法は S503 と同様である。S601 の処理により、二次元コード各辺における境界線上の位置が検出される。

#### 【 0 0 7 7 】

図 1 2 の二次元コード領域の例では、図 1 4 のように、二次元コード上側の境界線上に A1 ～ A7、下側の境界線上に B1 ～ B7、左側の境界線上に C1 ～ C5、右側の境界線上に D1 ～ D5 の位置がそれぞれ検出される。

#### 【 0 0 7 8 】

S602 ～ S605 では、上側、下側、左側、右側の検出位置を直線近似し、それぞれの傾き角度  $a_1 \sim a_4$  を求める。図 1 4 の例では、上側については A1 ～ A7 を通る近似直線 A の傾き  $a_1$  を算出する。下側、左側、右側に関しても同様にして傾き角度を算出する。

#### 【 0 0 7 9 】

S606 では、上側と下側の傾き角度に対して左側と右側の傾き角度が直行することを利用し、 $a_1 \sim a_4$  の平均の傾きを計算して二次元コードの傾き角度  $\theta$  を次式により決定する。

#### 【 0 0 8 0 】

$$\theta = (a_1 + a_2 + (a_3 - 90) + (a_4 - 90)) \div 4$$

S607 では、 $\theta$  を所定の値  $\theta_{\min}$  と比較し、 $\theta$  が  $\theta_{\min}$  より大きい場合には S608 で傾き補正処理を行って終了し、 $\theta$  が  $\theta_{\min}$  以下の場合には傾き補正を行わずに処理を終了する。これにより、傾き角度が小さく、傾き補正の必要のない場合には回転処理を行わないことで、処理時間を短縮できる。

#### 【 0 0 8 1 】

読取方向を検出して正立像に自動補正する方法には、公知例として「画像状態検出装置及びそれを利用した文書画像補正装置」（特許公開平成 11 年第 316

798号)がある。

【0082】

図3のS101の処理を行う前に、この補正処理を行って入力画像を正立像に補正しておくことにより、S101以降の処理はこの正立像に補正した画像に対して行われるため、回転や鏡像反転が補正された二次元コードが抽出される。

図15は、二次元コードを構成するブロックを検出するためのその他の方法を説明するための図面であり、8×8画素のブロックにおいて着目画素に対する黒画素間距離の求め方を説明するための図である。ここで、図示左上の画素をX1、1とし、X1、1の右側の画素をX2、1、X1、1の下側の画素をX2、1のように添え字を変えて各画素を表す。

【0083】

まず、図5の黒画素X4、4に着目し、X4、4に対して上下左右それぞれの方向で最も近い黒画素がX4、3、X4、8、X1、4、X6、4であった場合、X4、4についての黒画素間の距離D4、4は次式で算出する

$$D_{4,4} = |X_{4,4} - X_{4,3}| + |X_{4,4} - X_{4,8}| + |X_{4,4} - X_{1,4}| + |X_{4,4} - X_{6,4}| \dots \quad (1)$$

なお、上下左右のうち、黒画素が存在しない方向があった場合や、着目画素がブロック外縁にあり画素が存在しない方向がある場合には、該方向に相当する項はゼロとする。例えば、X1、4を含めX4、4の左側が全て白画素であった場合には、X4、4についての黒画素間の距離は次式のようにになる。

【0084】

$$D_{4,4} = |X_{4,4} - X_{4,3}| + |X_{4,4} - X_{4,8}| + |X_{4,4} - X_{6,4}| \dots \quad (2)$$

一般に、着目した黒画素についての黒画素間隔は次式により算出する。

【0085】

$$D_{i,j} = |X_{i,j} - X_{i,j-a}| + |X_{i,j} - X_{i,j+b}| + |X_{i,j} - X_{i-c,j}| + |X_{i,j} - X_{i+d,j}|$$

$$= a + b + c + d \quad \cdots (3)$$

式(3)の第一項から第四項は、それぞれ上下左右にa画素、b画素、c画素、d画素離れた黒画素との距離を求めることを意味するため、着目画素 $X_{i,j}$ についての黒画素間隔は $a + b + c + d$ と表せる。

## 【0086】

ブロック内の全ての黒画素についての黒画素間の平均距離Dは、ブロック内の黒画素数をNとして、次式により算出することができる。

## 【0087】

$$D = (\sum \sum D_{i,j}) / N \quad \cdots (4)$$

ここで、最初の $\sum$ は $D_{i,j}$ を $i=1$ からNまでの加算すること、2個目の $\sum$ は $D_{i,j}$ を $j=1$ からNまで加算することを示す。

図16は、文字ブロック(A)と二次元コードブロック(B)それぞれにおける黒画素分布の例である。一般的に、文字ブロックでは黒画素が局所的に集まって存在し、二次元コードブロックでは黒画素が比較的ばらばらな位置に存在している。

## 【0088】

図17は、図16に図示される文字ブロックと二次元コードブロックについて、前述した画素間の距離を算出した例である。太線にて描かれた小さい矩形は黒画素を表し、矩形内の数値は黒画素について式(1)により黒画素間を算出した値である。

## 【0089】

図16の例では、文字ブロック(A)の黒画素間の合計は63、ブロック内の黒画素数は24個なので平均黒画素間は $D = 2.6 (63 \div 24)$ となる。一方、二次元コードブロック(B)の黒画素間の合計は131、ブロック内の黒画素数は27なので平均黒画素間は $D = 4.9 (131 \div 27)$ となる。二次元コードブロックの平均黒画素間のほうが文字ブロックよりも大きい。

## 【0090】

平均黒画素間Dは、式(3)からも明らかなように黒画素同士の間隔が大きい

ほど値が大きくなる。図 1 6 に示したように、二次元コードブロック (A) 内の黒画素は、文字ブロック (B) に比較してばらばらな位置に存在しているため、各黒画素の間隔が文字ブロックよりも大きくなり、従って、二次元コードブロックの平均黒画素間  $D$  も文字ブロックよりも大きな値になる。この黒画素間隔の違いを利用してブロックが文字の一部なのか二次元コードの一部なのかを識別する。

#### 【0091】

文字ブロックと二次元コードを識別する黒画素間の閾値  $D_{th}$  は次のように決める。まず、文字のみを含む複数の文書画像をブロックに分割し、全てのブロックについて求めた平均黒画素間の平均  $D_1$  を求める。次に、複数の二次元コード画像をブロック分割し、文字ブロックの場合と同様に全ブロックの平均黒画素間の平均  $D_2$  を求める。こうして求めた  $D_1$ 、 $D_2$  はそれぞれ文字および二次元コードブロックの平均的な平均黒画素間を表すと考えられるので、 $D_1$  と  $D_2$  の中点を閾値  $D_{th}$  に決定する。

#### 【0092】

ブロックの平均黒画素間が  $D_{th}$  未満ならば文字ブロック、 $D_{th}$  以上ならば二次元コードブロックと識別することができる。

#### 【0093】

##### 【発明の効果】

以上説明したように、本発明の二次元コード抽出方法によれば、二次元コードと文字や図表が混在する画像データの中から二次元コードだけを自動的に抽出することができる。このため、二次元コードを文書・図表と共に印刷した紙面において、イメージスキャナにより二次元コードを含むように読み取りを行なうと、読み取った画像から二次元コードが自動抽出されるため、元のデータに正確に復元することができ、二次元コードの読み取りやすさの向上に寄与するところが多い。

##### 【図面の簡単な説明】

【図 1】 イメージスキャナで読み取られた二次元コードと文書／図表が混在する文書画像を模式的に図示した図面であり、 $M \times N$  のブロック単位で走査された状



態を示す。

【図 2】 図 1 で二次元コードとして検出されたブロックを拡大した図。

【図 3】 本発明の一実施形態による二次元コード検出手順を説明したフローチャート。

【図 4】 ブロック情報テーブルを示した図面。

【図 5】 二次元コードブロックの検出処理手順を示すフローチャート。

【図 6】 隣接ブロックにより構成される領域の検出処理手順を示すフローチャート。

【図 7】 二次元コード抽出を説明する図面。

【図 8】 ブロック情報テーブルの例を示す図。

【図 9】 二次元コード領域の検出処理を示すフローチャート。

【図 1 0】 検出された二次元コード領域の様子を示す図。

【図 1 1】 二次元コード良医の検出処理を示すフローチャート。

【図 1 2】 図 1 0 で検出された二次元コード領域を拡大した図面。

【図 1 3】 二次元コードの傾き検出とその補正処理を示すフローチャート。

【図 1 4】 傾いた二次元コードの様子を示す図。

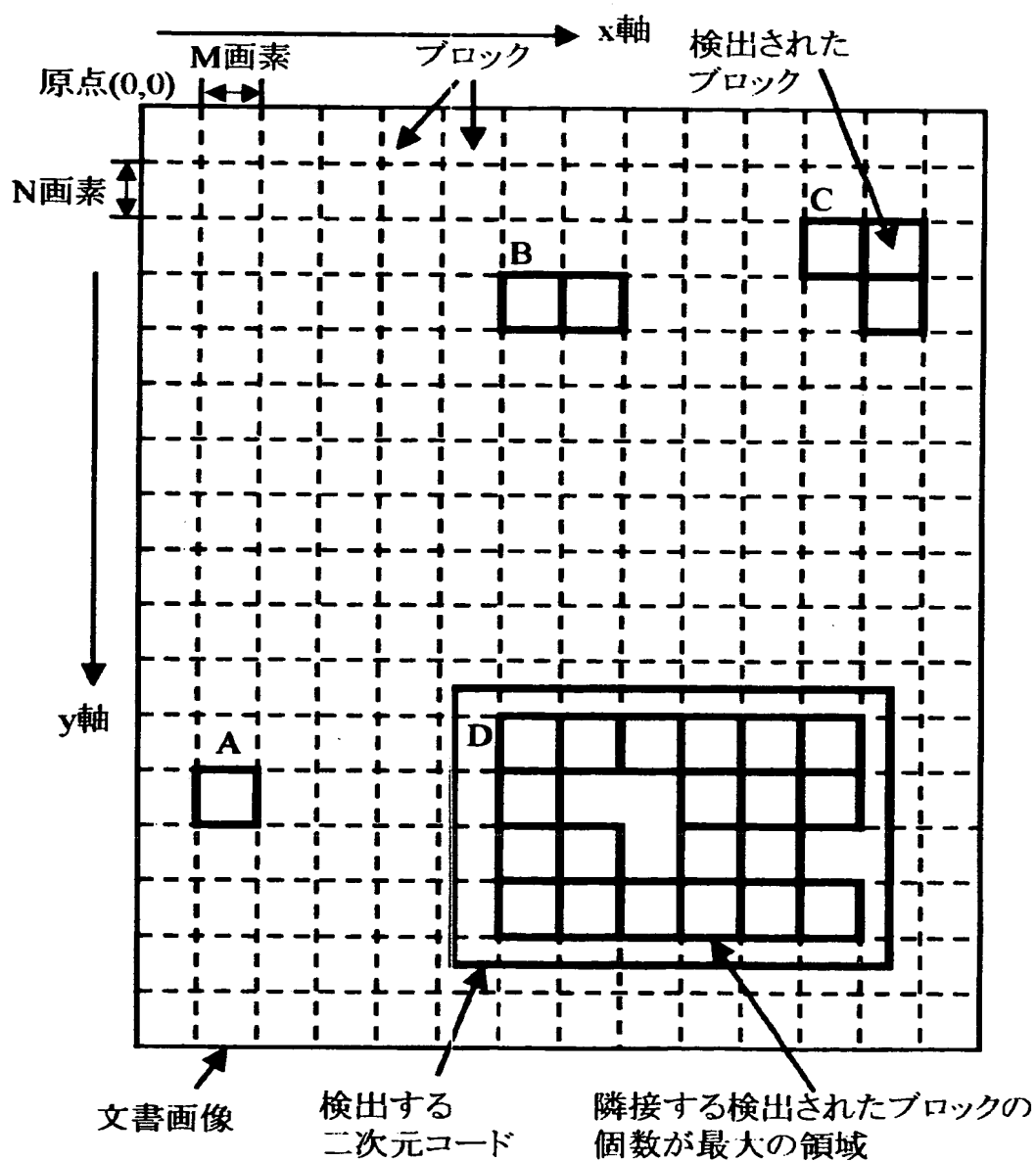
【図 1 5】 二次元コード領域検出を説明する図。

【図 1 6】 文字ブロックと二次元コードブロックの黒画素分布を示す図。

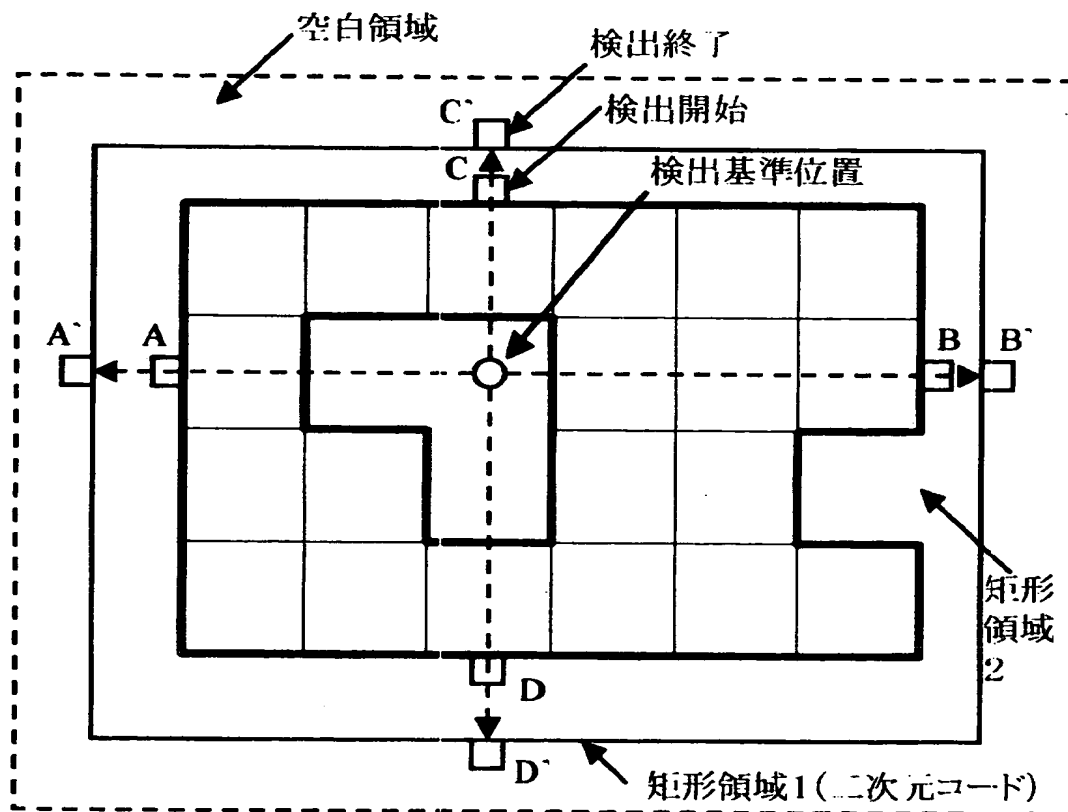
【図 1 7】 図 1 6 に図示された文字ブロックと二次元コードブロックとの黒画素間距離を算出した結果を示す図。

【書類名】 図面

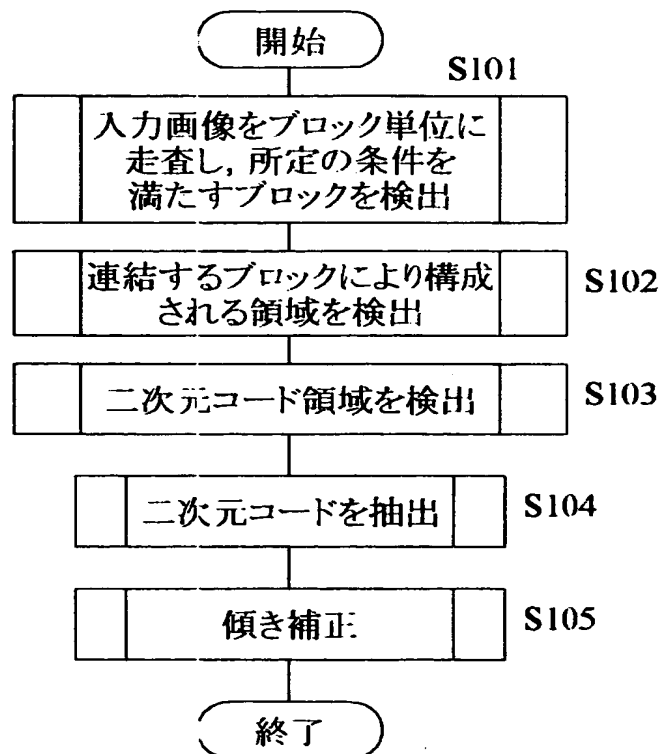
【図 1】



【図 2】



【図 3】

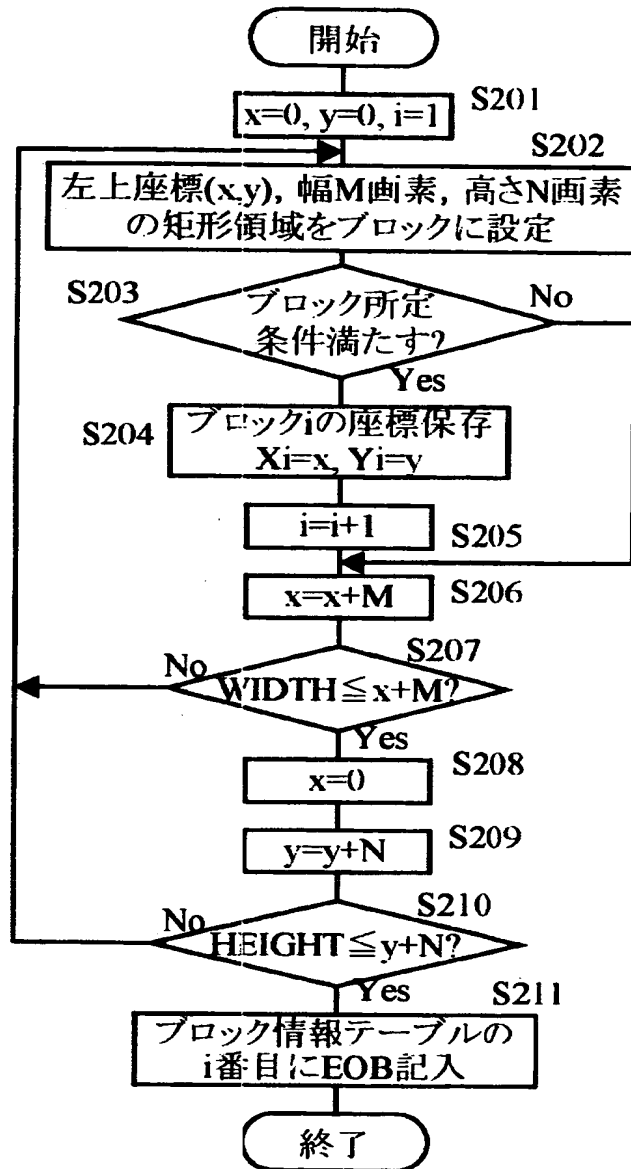


【図 4】

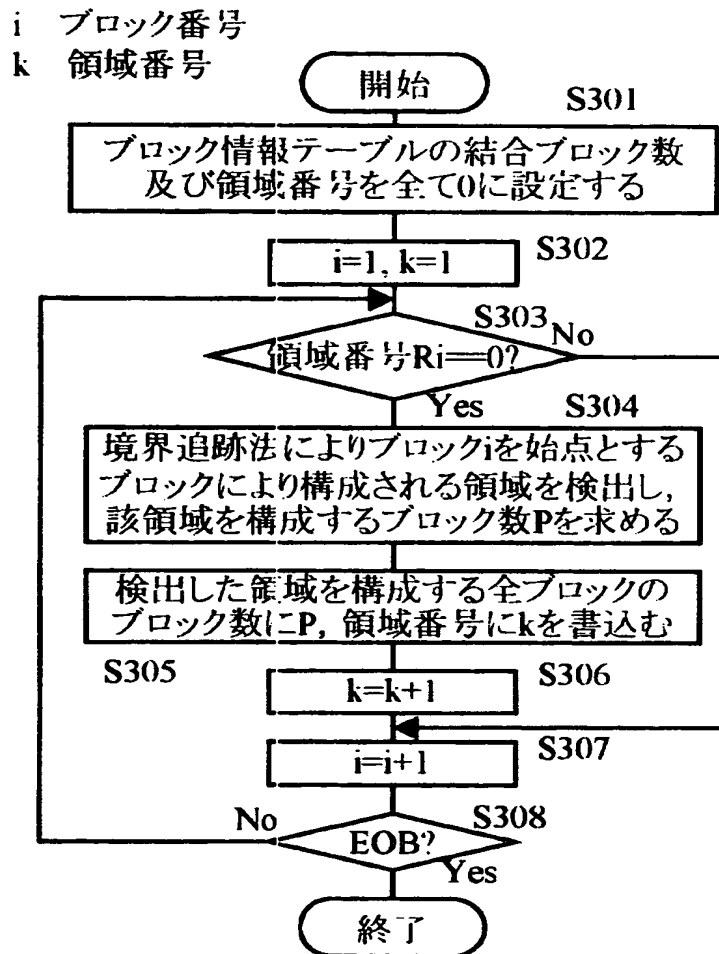
ブロック 番号	X座標	Y座標	結合ブ ック数	領域番号
1	X1	Y1	C1	R1
2	X2	Y2	C2	R2
...	...	...	...	...
i	Xi	Yi	Ci	Ri
...	...	...	...	...
n	Xn	Yn	Cn	Rn
EOB	—	—	—	—

EOB: End Of Block table,  
テーブルの終了を示す識別子

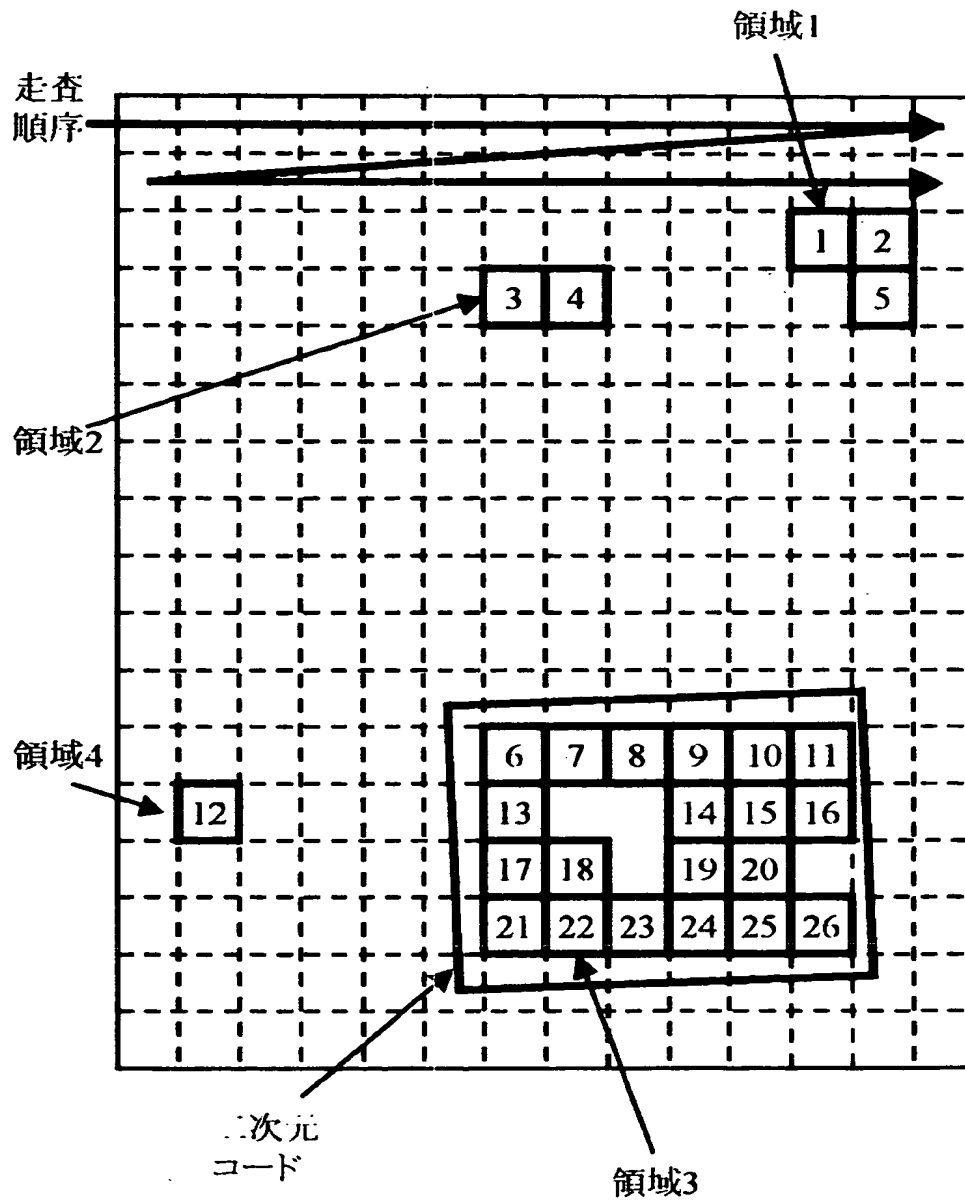
【図 5】



【図 6】



【図 7】

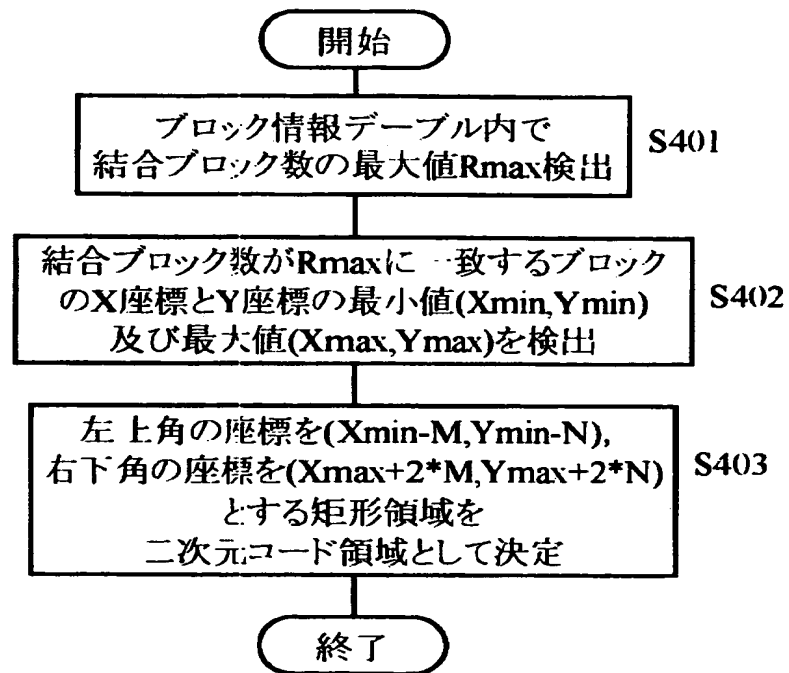




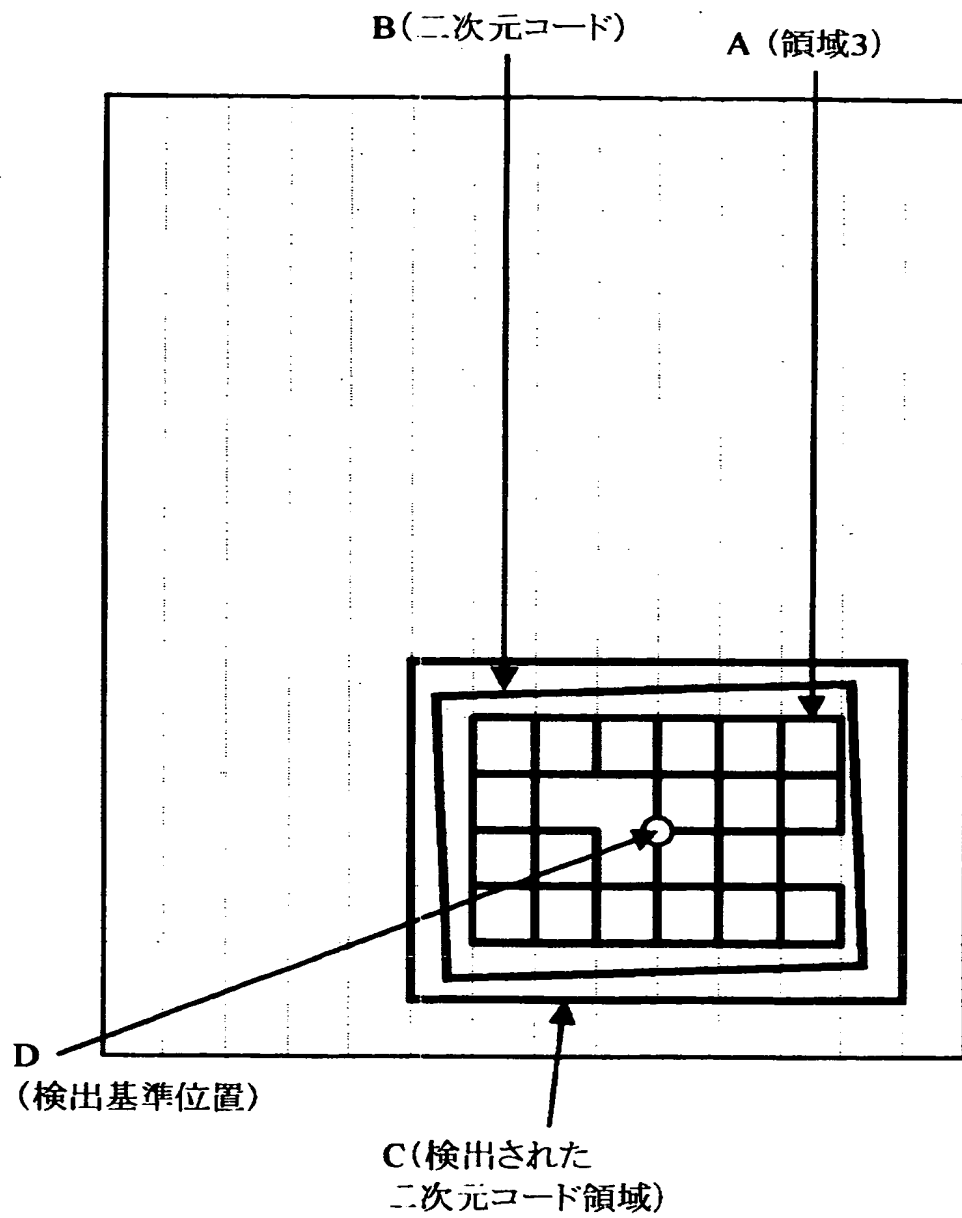
【図 8】

ブロック 番号	X座標	Y座標	結合 ブロック数	領域番号
1	352	64	3	1
2	384	64	3	1
3	192	96	2	2
4	224	96	2	2
5	384	96	3	1
6	192	352	20	3
7	224	352	20	3
8	256	352	20	3
9	288	352	20	3
10	320	352	20	3
11	352	352	20	3
12	32	384	1	4
13	192	384	20	3
...	...	...	...	...
26	352	448	20	3
EOB	—	—	—	—

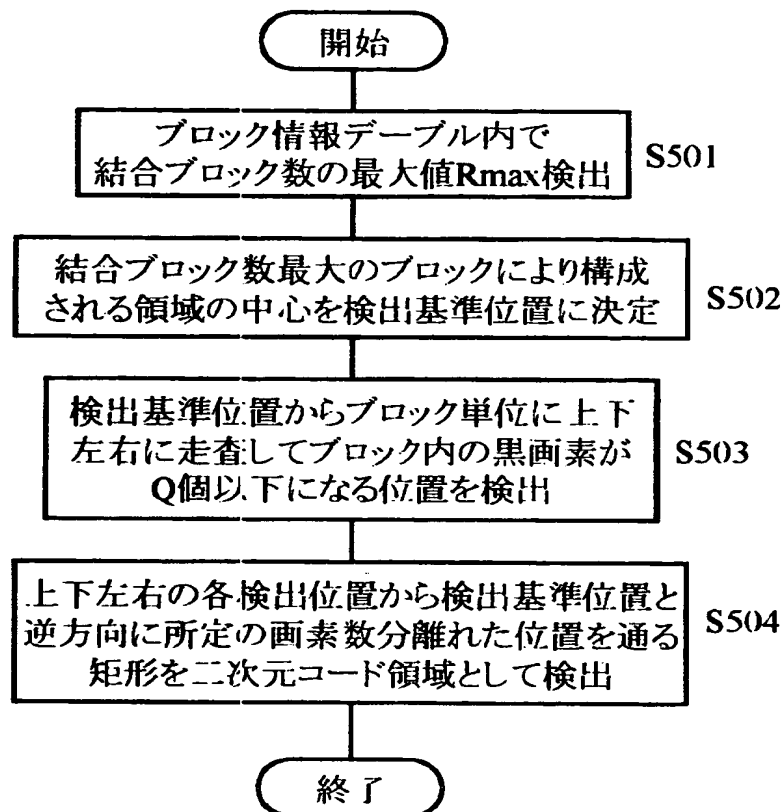
【図 9】



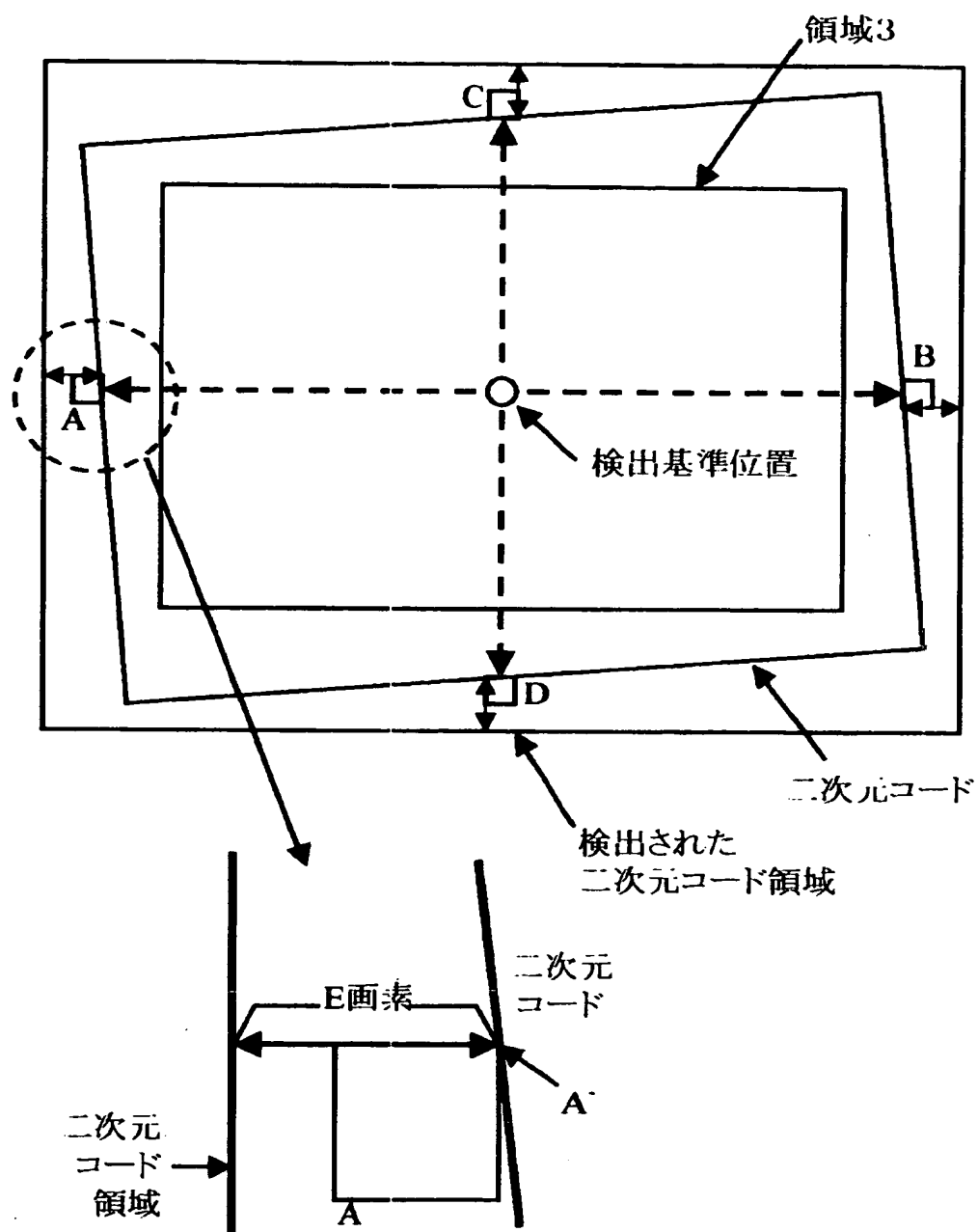
【図 1 0】



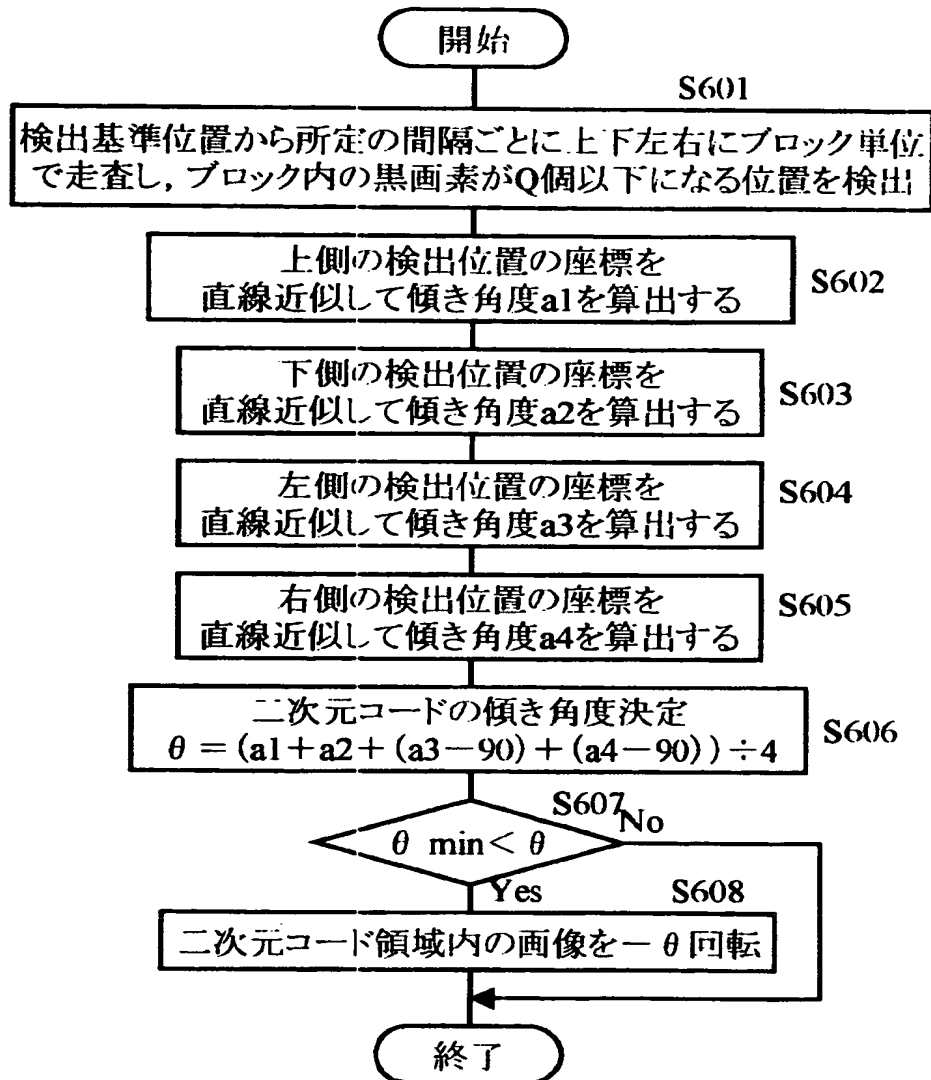
【図 1 1】



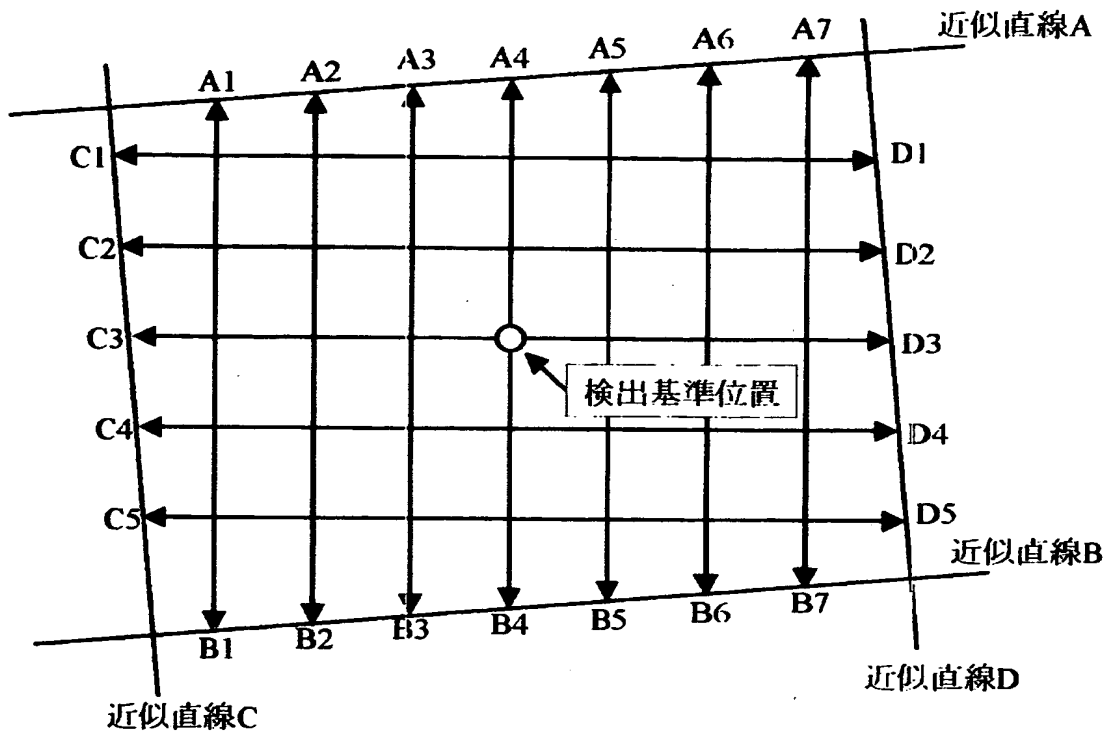
【図 1 2】



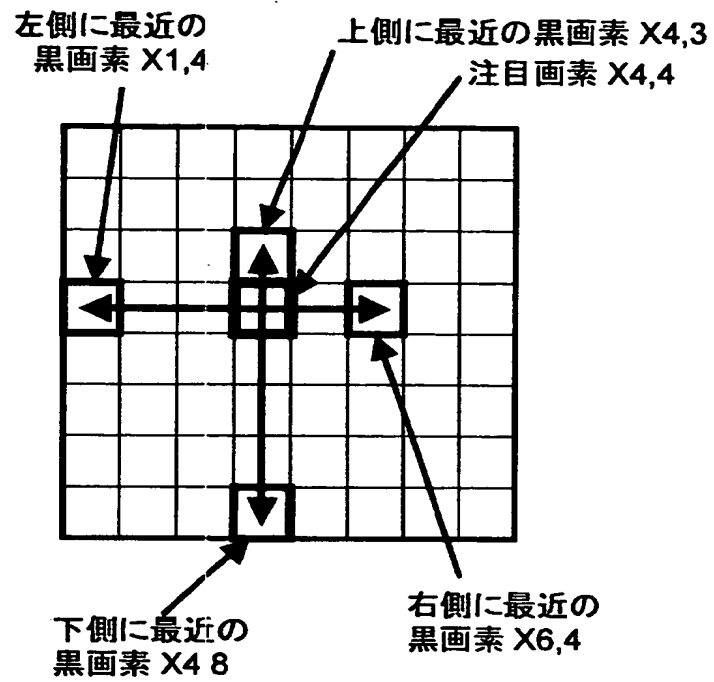
【図 1 3】



【図 1 4】

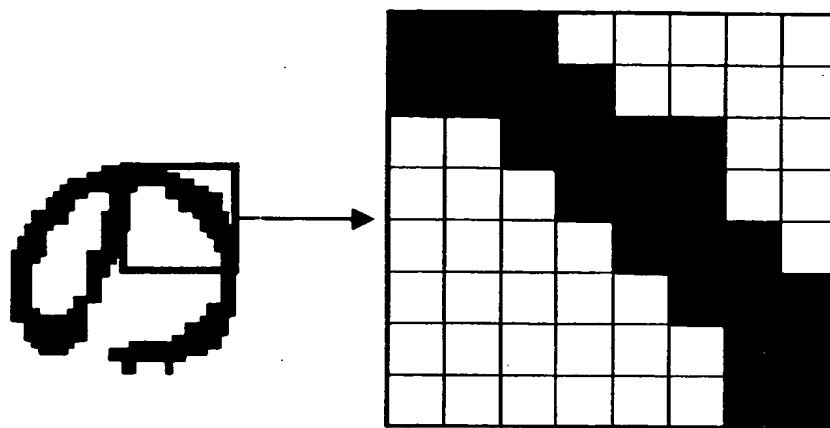


【図 1 5】

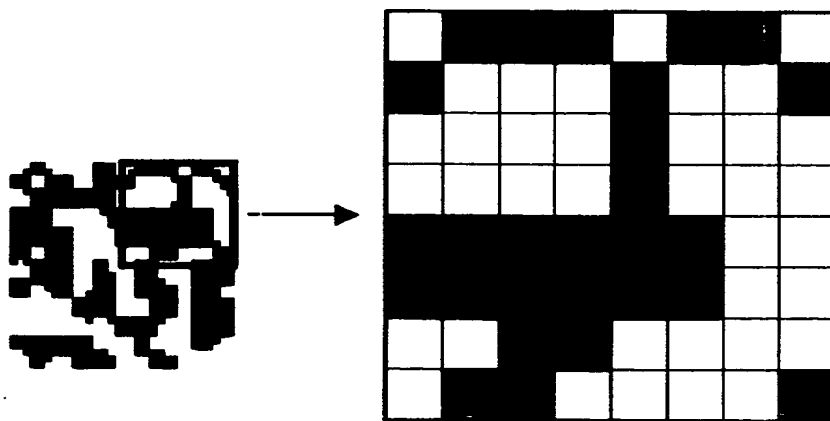




【図 1 6】



(A) 文字ブロック内の黒画素分布



(B) 二次元コードブロック内の黒画素分布

【図 17】

2	3	2					
2	2	4	2				
		2	4	3	2		
			2	4	3		
				2	4	2	
					2	4	2
						3	3
						2	2

$$D_{avg} = 63 / 24 = 2.6$$

文字ブロックの平均黒画素間

	5	6	6		6	1	
8				8			9
				2			
				2			
5	6	6	6	4	6		
1	5	4	4	3	2		
		3	2				
	3	7					11

$$D_{avg} = 131 / 27 = 4.9$$

二次元コードブロックの平均黒画素間

【書類名】 要約書

【要約】

【目的】 本発明は、読み取られた画像から二次元コードを抽出する二次元コード抽出方法に関し、文字やその他の画像が含まれた文書から二次元コードをより確実に抽出することを目的とする。

【構成】 読み取られた画像を $M \times N$ （ともに正の整数）単位で走査して、その領域毎に黒画素分布等を求めて、二次元コードと思われるブロック（A～D）を検出する。そして、検出されたブロックが連続している領域（D）を検出し、この領域を二次元コード領域として抽出する。

【選択図】 図 1

出 願 人 履 歷 情 報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日  
[変更理由] 住所変更  
住 所 神奈川県川崎市中原区上小田中4丁目1番1号  
氏 名 富士通株式会社